

10/069987

#2

PCT/JP00/06070

JP00/6070
4
日 本 国 特 許 庁
PATENT OFFICE
JAPANESE GOVERNMENT

06.09.00

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日
Date of Application:

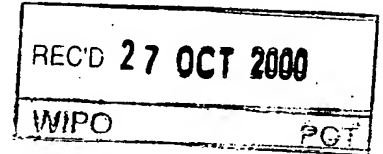
1999年 9月 8日

出 願 番 号
Application Number:

平成11年特許願第254149号

出 願 人
Applicant(s):

関 一

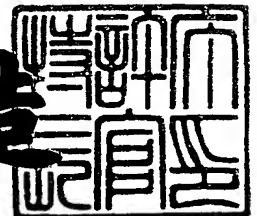


**PRIORITY
DOCUMENT**
SUBMITTED OR TRANSMITTED IN
COMPLIANCE WITH RULE 17.1(a) OR (b)

2000年10月13日

特許庁長官
Commissioner,
Patent Office

及 川 耕 造



出証番号 出証特2000-3083140

【書類名】	特許願
【整理番号】	RR01
【あて先】	特許庁長官殿
【国際特許分類】	G06F 9/30
【発明者】	
【住所又は居所】	愛媛県松山市道後喜多町4番38号
【氏名】	関 一
【特許出願人】	
【識別番号】	598003070
【住所又は居所】	愛媛県松山市道後喜多町4番38号
【氏名又は名称】	関 一
【手数料の表示】	
【予納台帳番号】	057509
【納付金額】	21,000円
【提出物件の目録】	
【物件名】	明細書 1
【物件名】	図面 1
【物件名】	要約書 1
【プルーフの要否】	要

【書類名】 明細書

【発明の名称】 レジスタ・リネーミング方式

【特許請求の範囲】

【請求項 1】

アーキテクチャ上規定されている論理レジスタの数を上回る数の物理レジスタと、割り付けられていない物理レジスタの番号を保持するフリー・リストと、各論理レジスタにそれぞれ対応して設けられた各エントリに物理レジスタ番号が書き込まれるようになっているマッピング・テーブルとを具備するout-of-order実行の可能なスーパースカラ・アーキテクチャを採用するプロセッサにおいて、

同時にデコードされる複数の命令に関して、

(a) 論理レジスタ番号を指定するデスティネーション・オペランドの各々に対し命令順にラベル付けし、

(b) 2 番目以降に位置する命令の各々において、各ソース・オペランドを該命令以前に位置する各命令のデスティネーション・オペランドと比較し、一致するものがあればそれと同じラベルを付し、

一致するものが複数個存在する場合には、最も後方に位置する命令におけるラベルを付し、

上記の処理の後に、

(a) ラベル付けされたオペランドについては、付されたラベルに対応して該フリー・リストから取り出される物理レジスタ番号を割り当て、ラベル付けされていない論理レジスタ番号を指定するオペランドについては、該マッピング・テーブルを論理レジスタ番号でアクセスして得られる物理レジスタ番号を割り当て、

(b) 該マッピング・テーブルにおいて、論理レジスタ番号を指定する各デスティネーション・オペランドで示されるエントリの内容を割り当てられる物理レジスタ番号に書き換え、

同じ論理レジスタ番号が複数の命令のデスティネーション・オペランドで指定されている場合には、最も後方に位置する命令において割り当てられる物理レジスタ番号に書き換えるようなレジスタ・リネーミング方式。

【請求項 2】

1 クロック・サイクル当りデコード／発行される命令の数だけの各々 1 つの物理レジスタ番号が書き込まれるようになっている F レジスタと、各々のエントリに 1 つの物理レジスタ番号が書き込まれるようになっているレジスタ・ファイルで構成される循環型の F I F O キューを具備し、

割り付けが解除された物理レジスタの番号を随時該キューの末尾に加え、

F レジスタから物理レジスタ番号が取り出されるのに応じて、物理レジスタ番号が取り出された F レジスタに該キューの先頭から物理レジスタ番号を補充するようになっているフリー・リスト。

【発明の詳細な説明】

【0 0 0 1】

【産業上の利用分野】

本発明は、1 クロック・サイクルで複数の命令を実行できるスーパースカラ・アーキテクチャをとるプロセッサにおいて、1 サイクル当り多くの命令のデコード／発行を可能とするようなレジスタ・リネーミング方式に関するものである。

【0 0 0 2】

【従来の技術】

Out-of-order 実行の可能なスーパースカラ・アーキテクチャをとるプロセッサにおいては、アーキテクチャ上規定されている論理レジスタの数を上回る数の物理レジスタを備えており、動的レジスタ・リネーミングという手法が活用される。

【0 0 0 3】

すなわち、各サイクルごとに、デコード中の全ての命令が使う論理レジスタ番号を比べて依存関係を調べ、その依存関係が解消するようにそれぞれの命令の各オペランドに物理レジスタ番号を割り当てる。

【0 0 0 4】

レジスタ・リネーミングを実現するには、リオーダ・バッファを用いる方式とマッピング・テーブルを用いる方式が知られている。リオーダ・バッファを用いる方式は、例えば、米国カリフォルニア州サンタクララ (Santa Clara) のインテル社 (Intel Corporation) から入手可能な商標ペンティアム (Pentium) プロセッ

サにおいて採用されている。マッピング・テーブルを用いる方式は、Keller, R. M. "Look-Ahead Processors." Computing Surveys, Vol. 7, no. 4 (December 1975), pp. 177-195 に示されており、例えば、米国カリフォルニア州マウンテンビュー (Mountain View) のM I P S社 (MIPS Technologies, Inc.) から入手可能な商標R10000プロセッサにおいて採用されている。

【0005】

リオーダ・バッファを用いる方式と比較して、マッピング・テーブルを用いる方式は、マッピング・テーブルをアクセスするためのステージが余分に必要となるが、命令間の依存性を調べる論理回路は単純なものですむ。

【0006】

【発明が解決しようとする課題】

スーパースカラ・プロセッサにおいては、より効率的に命令のデコード／発行を行い、リザベーション・ステーションと呼ばれる発行済みの命令を保持するバッファにより多くの命令をため込むことによって、処理性能を向上させることが求められている。

【0007】

しかしながら、1サイクル当り同時にデコード／発行できる命令の数を現在の水準よりさらに増やすのは、レジスタ・リネーミングのための論理回路が極度に複雑なものとなるため困難であるという問題がある。

【0008】

本発明は、上記問題点を解決するため創案されたものであり、その目的は、1サイクル当り同時にデコード／発行できる命令の数を現在の水準よりさらに増やすことを可能とするレジスタ・リネーミング方式を提供することにある。

【0009】

【課題を解決するための手段】

本発明によるレジスタ・リネーミング方式を採用するプロセッサは、フリーな、即ち、割り付けられていない物理レジスタの番号を保持するフリー・リスト、及び、各論理レジスタにそれぞれ対応して設けられた各エントリに物理レジスタ番号が書き込まれるようになっているマッピング・テーブルを備える。

【0010】

本発明によるレジスタ・リネーミング方式は、命令間の依存性の検出と物理レジスタ番号の割り当てを分離しパイプライン化することによって、1サイクル当りより多くの命令を同時にデコード／発行できるようにするものである。

【0011】

まず、同時にデコードされる複数の命令について以下のような処理が行われる。

(a) 論理レジスタ番号を指定するデスティネーション・オペランドの各々に対し、命令順にラベル付けする。このラベルは将来フリー・リストから取り出され割り当てられる物理レジスタ番号に置き換えられるべき指標である。

(b) 2番目以降に位置する命令の各々において、各ソース・オペランドがその命令以前に位置する各命令のデスティネーション・オペランドと比較され、一致するものがあればそれと同じラベルが付される。一致するものが複数個存在する場合には、最も後方に位置する命令におけるものを選択する。

【0012】

上記のような処理の後に、物理レジスタ番号の割り当て、及びマッピング・テーブルの更新が以下のように行われる。

(a) ラベル付けされているオペランドについては、付されたラベルに対応してフリー・リストから取り出される物理レジスタ番号を割り当てる。ラベル付けされていない論理レジスタ番号を指定するオペランドについては、マッピング・テーブルを論理レジスタ番号でアクセスして得られる物理レジスタ番号を割り当てる。

(b) マッピング・テーブルにおいて、論理レジスタ番号を指定する各デスティネーション・オペランドで示されるエントリの内容を割り当てられる物理レジスタ番号に書き換える。同じ論理レジスタ番号が複数の命令のデスティネーション・オペランドで指定されている場合には、最も後方に位置する命令において割り当てられる物理レジスタ番号に書き換える。

【0013】

以上のように、レジスタ・リネーミングを2段階で行う、即ちパイプライン化

することによって、比較的単純な回路を用いて、1サイクル当りより多くの命令を同時にデコード／発行することが可能となる。

【0014】

【実施例】

以下に、本発明によるレジスタ・リネーミング方式を実現する一実施例について、図面を参照しながら説明する。なお、以下に述べる本発明によるレジスタ・リネーミング方式を採用するプロセッサの実施例は、アーキテクチャ上はR0, R1, ... R7の8つの論理レジスタの存在を想定しているが、ハードウェアとしてはP00, P01, ... P31の32個の物理レジスタを備えるRISCマシンであり、1サイクル当り4つまでの命令を同時にデコード／発行できるように構成されているものとする。

【0015】

図1は本発明のレジスタ・リネーミング方式を実現するために必要となるハードウェアのブロック図であって、1はマッピング・テーブル、2はフリー・リストを表している。

【0016】

次に、上記各構成要素ごとにその詳細な構成を説明する。

【0017】

(A) マッピング・テーブル (MT)

マッピング・テーブル (以下ではMTと記す) は、デコード時点の論理レジスタと物理レジスタとの対応関係を保持するものである。

【0018】

図1に示されるように、本実施例においては、MT1はR0, R1, ... R7の8つの論理レジスタにそれぞれ対応して設けられた10～17の8つのレジスタで構成される。MTを構成する各々のレジスタは、1つの物理レジスタ番号が書き込まれる構成となっている。

【0019】

(B) フリー・リスト (FL)

フリー・リスト (以下ではFLと記す) は、フリーな、即ち、割り付けられて

いない物理レジスタの番号を保持するものである。

【 0 0 2 0 】

初期状態においては、全ての物理レジスタの番号が F L に登録されている。フリーな物理レジスタを割り付ける際に、F L からフリーな物理レジスタの番号が取り出される。逆に、ある物理レジスタの割り付けが解除されれば、その物理レジスタの番号が F L に登録されるようになっている。

【 0 0 2 1 】

図 1 に示されるように、本実施例においては、F L 2 は F レジスタ 2 1、2 2、2 3、2 4 とレジスタ・ファイル 2 8 を具備する。

【 0 0 2 2 】

各々の F レジスタは 1 つの物理レジスタ番号が書き込まれる構成となっており、F L からフリーな物理レジスタの番号が取り出されるのは、F レジスタからとなっている。即ち、1 サイクル当りデコード／発行される命令の数だけ F レジスタが必要で、本実施例のプロセッサの場合、F1、F2、F3、F4 の 4 つの F レジスタを備えている。

【 0 0 2 3 】

F L にあるレジスタ・ファイル 2 8 は、各々のエントリに 1 つの物理レジスタ番号が書き込まれるようになっており、循環型の F I F O キューを構成する。即ち、割り付けが解除された物理レジスタの番号がキューの末尾に加えられ、物理レジスタ番号が取り出された F レジスタにキューの先頭から物理レジスタ番号が補充されるようになっている。

【 0 0 2 4 】

ついで、本実施例におけるレジスタ・リネーミングの過程を説明する。

【 0 0 2 5 】

本実施例においては、①命令間の依存性の検出、②物理レジスタ番号の割り当て及び M T の更新、の 2 ステージで、レジスタ・リネーミングが行われる。以下に各ステージごとに動作内容を説明する。

【 0 0 2 6 】

① 命令間の依存性の検出

同時にデコードされる複数の命令について以下のような処理が行われる。

(a) 論理レジスタ番号を指定するデスティネーション・オペランドの各々に対し、@1, @2, ...と命令順にラベル付けする。本実施例においては、ラベル@1, @2, @3, @4は、FLにあるFレジスタF1, F2, F3, F4にそれぞれ対応するものである。

(b) 2番目以降に位置する命令の各々において、各ソース・オペランドがその命令以前に位置する各命令のデスティネーション・オペランドと比較され、一致するものがあればそれと同じラベルが付される。一致するものが複数個存在する場合には、最も後方に位置する命令におけるものを選択する。

【0027】

② 物理レジスタ番号の割り当て及びMTの更新

(a) ラベル付けされているオペランドについては、付されたラベルに対応してFLから取り出される物理レジスタ番号を割り当てる。ラベル付けされていない論理レジスタ番号を指定するオペランドについては、MTを論理レジスタ番号でアクセスして得られる物理レジスタ番号を割り当てる。

(b) MTにおいて、論理レジスタ番号を指定する各デスティネーション・オペランドで示されるエントリの内容を割り当てられる物理レジスタ番号に書き換える。同じ論理レジスタ番号が複数の命令のデスティネーション・オペランドで指定されている場合には、最も後方に位置する命令において割り当てられる物理レジスタ番号に書き換える。

【0028】

以上が、本実施例におけるレジスタ・リネーミングの過程であるが、ついで、具体的な動作例について説明する。いま、本実施例のプロセッサにおいて、以下のような4つの命令を同時にデコード／発行するものとしよう。

```
mul  R0, R1, R2    ; R0 = R1 * R2
mul  R1, R3, R4    ; R1 = R3 * R4
add  R0, R0, R1    ; R0 = R0 + R1
div  R0, R0, R5    ; R0 = R0 / R5
```

【0029】

以上の命令列は、それ以前に位置する命令が全て完了したと仮定すれば、R1, R2, R3, R4, R5の各レジスタがそれぞれデータD1, D2, D3, D4, D5を保持するものとして、 $\{(D1 \times D2) + (D3 \times D4)\} / D5$ を計算し、レジスタR0に格納する、というものである。

【0030】

本実施例のプロセッサにおいて、上記命令列に関してレジスタ・リネーミングが行われる過程を、以下に詳細に説明する。

【0031】

まず、命令間の依存性の検出を行い、レジスタ・リネーミングのための前処理が次のように行われる。

```
mul R0(@1), R1(none), R2(none)
```

```
mul R1(@2), R3(none), R4(none)
```

```
add R0(@3), R0(@1), R1(@2)
```

```
div R0(@4), R0(@3), R5(none)
```

【0032】

即ち、各命令において命令コードの次に示されるデスティネーション・オペランドに対して、それぞれ順にラベル@1, @2, @3, @4を付ける。

【0033】

2番目以降に位置する命令の各々において、各ソース・オペランドがその命令以前に位置する各命令のデスティネーション・オペランドと比較され、一致するものがあればそれと同じラベルが付される。一致するものが複数個存在する場合には、最も後方に位置する命令におけるものを選択する。

【0034】

2番目に位置する命令においては、各ソース・オペランドが1番目に位置する命令のデスティネーション・オペランドR0と比較され、この場合一致するものはないのでラベルは付けられない。

【0035】

3番目に位置する命令においては、各ソース・オペランドが1番目及び2番目に位置する命令のデスティネーション・オペランドと比較され、この場合、第1

ソース・オペランドが1番目に位置する命令のデスティネーション・オペランド、第2ソース・オペランドが2番目に位置する命令のデスティネーション・オペランドと一致するので、それぞれラベル@1, @2を付ける。

【0036】

4番目に位置する命令においては、各ソース・オペランドが1番目、2番目及び3番目に位置する命令のデスティネーション・オペランドと比較され、この場合、第1ソース・オペランドが1番目及び3番目に位置する命令のデスティネーション・オペランドと一致するので、後の方の3番目に対応するラベル@3を付ける。

【0037】

以上のようなレジスタ・リネーミングのための前処理は、適切な回路を用いることにより1サイクルで実現される。

【0038】

次に、物理レジスタ番号の割り当て及びMTの更新が行われるが、この時点のMT及びFLにある各Fレジスタの内容が図2(A)に示されるようなものであるとしよう。

【0039】

ラベル付けされているオペランドについては、付されたラベルに対応してFLから取り出される物理レジスタ番号を割り当てる。また、ラベル付けされていないオペランドについては、MTを論理レジスタ番号でアクセスして得られる物理レジスタ番号を割り当て、与えられた命令列は次のような内容に読み替えられる。

```
mul  P19, P03, P22
mul  P08, P29, P05
add  P27, P19, P08
div  P21, P27, P07
```

【0040】

また、MTにおいて、各命令のデスティネーション・オペランドで示されるエントリの内容を割り当てられる物理レジスタ番号に書き換える。ここでは、図2

(B)に示されるように、MTのR0及びR1に対応するレジスタがそれぞれP21、P08に書き換えられている。R0に関しては、P19、P27、P21の3重の割り当てが存在するが、MTの書き換えには最後のものが選ばれる。さらに、図2(B)に示されるように、物理レジスタ番号が取り出された各Fレジスタにフリーな物理レジスタの番号P31、P01、P17、P14が補充されている。

【0041】

以上が、本実施例のプロセッサにおける、与えられた命令列に関するレジスタ・リネーミングの過程である。

【0042】

【発明の効果】

以上のように、本発明のレジスタ・リネーミング方式によれば、命令間の依存性の検出と物理レジスタ番号の割り当てを分離しパイプライン化することによって、比較的単純な回路を用いて、1サイクル当たりより多くの命令を同時にデコード／発行することが可能となる。

【図面の簡単な説明】

【図1】

本発明のレジスタ・リネーミング方式を実現するために必要となるハードウェアのブロック図である。

【図2】

本発明の一実施例において、ある命令列に関してレジスタ・リネーミングが行われる際の、マッピング・テーブル及び各Fレジスタの内容の変化を具体的に示す図である。

【符号の説明】

- 1 マッピング・テーブル (MT)
- 2 フリー・リスト (FL)
- 10～17 マッピング・テーブル (MT) を構成するレジスタ
- 21～24 Fレジスタ
- 28 レジスタ・ファイル

【書類名】

図面

【図 1】

R0	- 10 -
R1	- 11 -
R2	- 12 -
R3	- 13 -
R4	- 14 -
R5	- 15 -
R6	- 16 -
R7	- 17 -

1 マッピング・テーブル

F1	- 21 -
F2	- 22 -
F3	- 23 -
F4	- 24 -
<div style="border: 1px solid black; height: 150px; display: flex; align-items: center; justify-content: center;"> <div style="text-align: center;">- 28 -</div> </div>	

2 フリー・リスト

【図 2】

(A)		(B)	
R0	P18		P21
R1	P03		P08
R2	P22		P22
R3	P29		P29
R4	P05		P05
R5	P07		P07
R6	P25		P25
R7	P12		P12
F1	P19		P31
F2	P08		P01
F3	P27		P17
F4	P21		P14

【書類名】 要約書

【要約】

【目的】 スーパースカラ・アーキテクチャをとるプロセッサにおいて、1サイクル当たり多くの命令のデコード／発行を可能とするようなレジスタ・リネーミング方式を提供する。

【構成】 割り付けられていない物理レジスタの番号を保持するフリー・リストと、各論理レジスタにそれぞれ対応して設けられた各エントリに物理レジスタ番号が書き込まれるようになっているマッピング・テーブルを備え、命令間の依存性の検出と物理レジスタ番号の割り当てを分離しパイプライン化することによって、1サイクル当たりより多くの命令を同時にデコード／発行するようになっている。

【選択図】 図 1

認定・付加情報

特許出願の番号	平成11年 特許願 第254149号
受付番号	59900874118
書類名	特許願
担当官	濱谷 よし子 1614
作成日	平成11年 9月13日

<認定情報・付加情報>

【提出日】

平成11年 9月 8日

出 願 人 履 歴 情 報

識別番号 [5 9 8 0 0 3 0 7 0]

1. 変更年月日 1 9 9 7 年 1 1 月 2 0 日

[変更理由] 新規登録

住 所 愛媛県松山市道後喜多町 4 番 3 8 号

氏 名 関 一

This Page Blank (uspto)